

CLAIMS

- 1 1. A method for processing an original graphical element having an associated original
2 type, the method comprising:
3 blending at least part of the original graphical element and at least part of at least one
4 other graphical element to produce a transformed graphical element having an associated
5 transformed type, the transformed type being different than the original type;
6 storing information about the original type for the original graphical element; and
7 processing at least one of the transformed graphical element and an adjacent graphical
8 element using the stored information about the original type.
- 1 2. The method of claim 1, further comprising:
2 storing information about a type associated with the at least one other graphical
3 element.
- 1 3. The method of claim 1, further comprising:
2 storing information about a colorspace and a color for the original graphical element.
- 1 4. The method of claim 1, further comprising:
2 storing an original shape of the at least part of the original graphical element.
- 1 5. The method of claim 4 wherein:
2 storing the original shape includes storing the original shape as a path of the at least
3 part of the original graphical element.
- 1 6. The method of claim 4 wherein:
2 storing the original shape includes storing the original shape as a text glyph of the
3 original graphical element.

- 1 7. The method of claim 4 wherein:
2 processing includes locating one or more edges in the transformed graphical element
3 using the stored original shape.
- 1 8. The method of claim 1 wherein:
2 the transformed graphical element is a rasterized representation of the blended at least
3 part of the original graphical element and at least part of the at least one other graphical
4 element.
- 1 9. The method of claim 1 wherein:
2 storing information about the original type includes storing information about the
3 original type in an invisible graphical element.
- 1 10. The method of claim 1 wherein:
2 storing information about the original type includes storing information about the
3 original type in an XML element.
- 1 11. The method of claim 1 wherein:
2 processing includes trapping at least one of the transformed graphical element and the
3 adjacent graphical element.
- 1 12. The method of claim 11 wherein:
2 trapping includes using a path of the transformed graphical element to represent a
3 path of the at least part of the original graphical element.
- 1 13. The method of claim 11 wherein:
2 trapping includes using a color of the transformed graphical element to calculate a
3 color of a trap element.

1 14. The method of claim 11 wherein:
2 trapping includes using trapping rules that depend on the stored information about the
3 original type.

1 15. The method of claim 1 wherein:
2 processing includes halftoning at least one of the transformed graphical element and
3 the adjacent graphical element.

1 16. The method of claim 1 wherein:
2 blending includes flattening at least part of the original graphical element and at least
3 part of the at least one other graphical element to produce the transformed graphical element.

1 17. The method of claim 1 wherein:
2 at least one of the original graphical element and the at least one other graphical
3 element is a transparent graphical element; and
4 the transformed graphical element is an opaque graphical element.

1 18. The method of claim 1 wherein:
2 the original graphical element was produced by blending two or more previous
3 graphical elements; and
4 storing information about the original type includes storing information about a type
5 associated with at least one of the previous graphical elements.

1 19. The method of claim 1 wherein:
2 the original type comprises a member of a set of types, the types in the set of types
3 including raster, vector stroke, vector fill, image mask, soft mask, glyph, and gradient.

1 20. The method of claim 1 wherein:
2 the transformed type is raster.

1 21. The method of claim 1 wherein:
2 the original type is not associated with the transformed graphical element.

1 22. A computer program product, tangibly stored on a computer-readable medium, for
2 processing an original graphical element having an associated original type, the product
3 comprising instructions operable to cause a programmable system to:

4 blend at least part of the original graphical element and at least part of at least one
5 other graphical element to produce a transformed graphical element having an associated
6 transformed type, the transformed type being different than the original type;

7 store information about the original type for the original graphical element; and

8 process at least one of the transformed graphical element and an adjacent graphical
9 element using the stored information about the original type.

1 23. The product of claim 22 wherein:

2 the transformed graphical element is a rasterized representation of the blended at least
3 part of the original graphical element and at least part of the at least one other graphical
4 element.

1 24. The product of claim 22 wherein:

2 the instructions operable to cause a programmable system to process include
3 instructions operable to cause a programmable system to trap at least one of the transformed
4 graphical element and the adjacent graphical element.

1 25. The product of claim 22 wherein:

2 at least one of the original graphical element and the at least one other graphical
3 element is a transparent graphical element; and

4 the transformed graphical element is an opaque graphical element.

1 26. The product of claim 22 wherein:

2 the original type is not associated with the transformed graphical element.